A constant flow of site traffic is every Webmaster's dream, but sometimes that steady stream can turn into an unexpected flood. Here's how to stay afloat.

managing traffic spikes

Kevin Savetz

Your Web servers are humming along peacefully, doling out Web pages at a leisurely rate as Internet users from around the globe request them. Then something happens that drives hordes of visitors to your site, and that manageable traffic flow turns into a crushing load. Your Web servers buckle. Visitors endure tedious waits for each page—or worse, "server unavailable" errors.

Lots of people are visiting your company's Web site—that's a good thing. But too much traffic can bog down your servers, clog your bandwidth, and frustrate site visitors.

Traffic spikes can be broken into two broad categories: the ones you expect, and the ones you don't. Spikes that you can anticipate—those on the day of a major product release or ad campaign, for instance—you can prepare for in advance. Even if you're caught unawares by a traffic spike, perhaps caused by a major news event or a link from Slashdot, you may be able to reduce its impact after the fact.

weathering a crisis

On September 11, 2001, news-hungry users from around the world flocked to CNN.com for the latest headlines about the terrorist attacks in New York City and Washington, D.C. On an average day, the site serves about forty million page views. On September 11, that number climbed to more than 162.4 million page views, then to 337.4 million the following day.

Spikes like those called for drastic measures. "The first step we took was to slim down the page. We took off the graphics and the pictures and kept the most relevant information out there," says CNN.com spokeswoman Elizabeth Barry. At 10 A.M., she explains, the site's homepage consisted of the CNN logo, a single image of the World Trade Center, and minimal text.

But streamlined content wasn't enough. Without additional hardware, CNN.com could never have handled its increased load.

Fortunately, as part of AOL Time-Warner, CNN.com could borrow servers from its sister Web sites. "We were able to increase our server capacity and bring in more servers to support the amount of users that were trying to log into the site," Barry says.

Once engineers had increased the site's server capacity, the homepage regained much of its usual look, including links and multiple images. Webmasters kept advertisements off the site for several days, however, in order to free on-screen real estate for news content.

Even so, CNN.com did all it could to reduce the load on its servers by routing visitors to alternate information channels. "Another thing we did was to increase the number of breaking news emails that we sent out," Barry explains. "That way people who couldn't get to the site could still get the information they needed." By sending news to users who had subscribed to their email list, CNN could deliver news as it happened, while decreasing the load on its Web server from users looking for the latest news.

Even if you're caught unawares by a traffic spike, you may be able to reduce its impact after the fact.

prepare for the crunch

CNN's example shows the importance of having a traffic management strategy in place. There's plenty that you can do to prepare for traffic spikes in advance, and to deal with them when they're a reality. Your options depend on your budget and timeframe. They may include adding hardware, employing a content caching system, or just making the most of the resources you already have.

Traffic spikes tax both server load and network bandwidth. It is important to know which one is the weak link during busy times—or if both are deficient. You can have plenty of server power, but without enough bandwidth, visitors won't be able to take advantage of it. Likewise, all the bandwidth in the world won't help you if your server can't keep up.

"We all have a tendency to try to fix our problems without even knowing what they are. It leads to a lot of wasted effort, money, and poor results," says Paul Froutan, vice president of engineering at managed hosting provider Rackspace (www.rackspace.com). "Figure out what the problem is, then look for a solution."

If it's server load that you're worried about, a simple but expensive approach is to add servers for more horsepower. "If you expect to get more traffic, add two, three, or four servers, and a load balancer," Froutan suggests. "All the servers contain the same content." When one server comes close to being overwhelmed, the load balancer passes requests to the next server.

Many load balancers offer more advanced features that network administrators might initially overlook, but can help to reduce load during peak access times. "Most newer load balancers can send different types of requests to different servers. So you may want to have a server just for JPEG images and a server just for media," Froutan says. Viewing a single Web page may mean the browser fetches content from several different servers, spreading the work around.

There's another category of hardware that can help ease the load on Web servers: cache servers. This is hardware that sits between the Web server and Internet connection, caching and distributing frequently accessed content to reduce the load on the server. "They can help in some situations, but they are basically just more servers," Froutan says. Smart load balancing and caching lets you be more refined about how you handle traffic. If you do it right, you'll need less horsepower.

Still other options fall into a category that Froutan calls "high-level refinements." For example, some vendors sell hardware that optimizes your connections to reduce the servers' overhead in setting up and breaking down TCP/IP connections. The Web server can send a single stream of information to the network server rather than numerous short connections directly to the user, improving connection times by 5 to 20 percent. This can be very effective for applications such as online surveys or test taking, where the user is logged in for several minutes. It is less effective for applications where the visitor connects, downloads content for five seconds, and doesn't talk to the server again. Generally, only critical, large-volume sites use these solutions, because of their high cost, which can reach $20,000 or higher.

configuration and site design

Adding expensive hardware isn't the only answer, nor should it be your first line of defense. Good site design and proper configuration of your Web servers can also help ease the workload during traffic spikes.

"Tweaking your Web application is very critical," Rackspace's Froutan says. The default configuration for a Web server may assume that the hardware it runs on has limited memory or CPU resources and constrain the number of simultaneous connections accordingly. With a fast processor and copious RAM, your application may run out of processes before your server is tapped out, creating an artificial performance ceiling.

"Setting limits properly can make a lot of difference. In the hands of a good system administrator, a well-configured Apache server can double the amount of content you can serve," Froutan says. "There's more freedom to tweak the server on Unix than Windows. It's probably more difficult to tweak on the Unix side because you have to know more, but you can get more out of it."

Another way to manage resources is to keep your visitors coming at a steady pace, instead of arriving in intermittent mobs that can swamp servers. Of course, you can't directly control when visitors will come, but you can control when new content appears on the site. By staggering the times when new content is made available, you can keep hits coming at an even pace. For example, rather than adding seven new articles in a single day, add one new feature each day of the week. Most content management tools will let Webmasters preload content for posting at predetermined times.

spread the traffic around

If you know that most users will want a particular piece of information from your site—a popular software download or a new press release, for example—you can reduce server load (and users' frustration) by making it available from the homepage, rather than burying it

several clicks down. If possible, make your most popular information available on third-party servers as well.

"The trick is to try to spread the traffic around," says Eric Kopf, information services manager at Aladdin Systems, makers of compression software. "With product launches or promotions, it's fairly easy to predict that traffic will spike for the first few weeks of a launch. So in anticipation of that, we try to post our files to a number of different sites that we can link to directly. This includes our own Web servers and our FTP servers, and various third-party servers.

Availability and cost are two competing issues in dealing with traffic spikes. "You have to have enough availability to service your users. You don't want to give them a bad experience when trying to get your software. But you have to balance this with your costs for providing this availability—especially when the product is free, such as shareware, demos, and updates," Kopf says.

Sometimes, striking that balance almost requires a crystal ball. Many third-party hosting sites want you to predict your bandwidth usage in advance and negotiate a fee based on that prediction. If you go over that limit, they may hit you with surcharges, so a traffic spike can cost a lot of money if your predictions are off. Not having a traffic spike when you expect one can be just as bad. "In a software distribution model, if the ship date slips, you are paying for a bandwidth level that you are not going to use. When you do ship, you are likely to go past your original traffic predictions and be hit with surcharges," Kopf says.

You can reduce the cost and frustration of badly timed spikes by using a hosting provider that doesn't lock you into tiers of projected bandwidth needs. If your provider will charge only for the bandwidth you actually use, then product release dates and resulting traffic spikes can be more flexible.

content distribution networks

Even the mightiest server farm may not be able to effectively do the job from one location, especially when you're trying to serve visitors from around the globe. A content distribution network can ease your servers' workload while delivering a huge speed boost to end users, by delivering content from servers that are geographically closer to them.

Various strategies are available for content distribution. One of the simplest is to use *mirrors*, servers that maintain complete copies of your Web site and automatically update when you make changes. Each visitor can click a link to choose the server that is closest.

A similar, more robust solution also uses mirrors, but automatically chooses the optimal server for each user. One example is Exodus Content Caching, a service the company acquired with its purchase of Digital Island earlier this year. Exodus's technology automatically sends visitors to the best server for them, depending on network traffic, server load, and geographic location.

A competing technology is Akamai's FreeFlow cache system. With it, your primary server doles out HTML pages, while images and other bandwidth-intensive content are served from Akamai's network of distributed cache servers.

What do you do when a throng of Internet users attacks your Web site, and you find that despite your best efforts, your server and bandwidth resources are maxed out?

If your ISP provides managed hosting, call and ask for a mirror server and a load balancer to distribute the work. It may take a few hours—and you'll pay for the privilege—but this can be an easy way to ramp up during a surprise attack.

If you simply don't have the hardware and bandwidth to service all the visitors during a traffic spike, cut your losses. "I would prefer to service some people and service them properly," Rackspace's Froutan says. "So I'd limit the number of connections and redirect everyone else so at least some people can get to me." You can redirect visitors to a simple text page that tells them the truth: "We're getting slammed, please try again later."

Another option is to drastically reduce the amount of work the server must do for each visitor by serving up fewer files. Remember that the first step CNN.com took on the morning of September 11, 2001, was to replace its normally elaborate and graphics-heavy home page with a simple, graphics-free page. This allowed the Web server to deliver news more quickly, by establishing fewer connections and less overhead for each visitor.

Think of preparing for traffic spikes like preparing for a flood: as long as you've got a disaster kit tucked away in the garage and the family has agreed on what to do during an emergency, things will probably go smoothly. You may not know exactly when the flood is coming or how deep the water will be, but you can be sure you have the resources to manage when it does. ◘

Kevin Savetz (www.savetz.com) has been a freelance computer technology journalist for a decade.

save bandwidth with compression

Compressed files take less time to transfer and use less bandwidth. The images that your Web server sends are compressed—doesn't it make sense to compress the HTML, too? Most modern Web browsers have the ability to accommodate this via the HTTP/1.1 protocol. Servers can compress files on the fly, or they can serve precompressed files; these options are known as dynamic and static compression.

Dynamic compression is necessary when serving dynamic Web pages, such as those built with PHP or ASP. But watch out—compressing these types of documents places a higher workload on the server, because it's forced to process a page every time it receives a request. Compression only really becomes beneficial when dealing with static pages that won't be updated between requests.

Not all Web servers include support for compression, so implementing it may require adding software to the server. Still, compression typically reduces the size of HTML, XML, and JavaScript files by 70 to 80 percent before transmission—a significant savings. Once implemented, visitors with compatible browsers will immediately enjoy faster access, and you'll save bandwidth.

—KS